A Multi-Service MAC Protocol in a Multi-Channel CSMA/CA for IEEE 802.11 Networks

Jalel Ben-Othman, Hind Castel, and Lynda Mokdad

Abstract: **The IEEE 802.11 wireless standard uses the carrier sense multiple access with collision avoidance (CSMA/CA) as its MAC protocol (during the distributed coordination function period). This protocol is an adaptation of the CSMA/CD of the wired networks. CSMA/CA mechanism cannot guarantee quality of service (QoS) required by the application because of its random access method. In this study, we propose a new MAC protocol that considers different types of traffic (e.g., voice and data) and for each traffic type different priority levels are assigned. To improve the QoS of IEEE 802.11 MAC protocols over a multi-channel CSMA/CA, we have developed a new admission policy for both voice and data traffics. This protocol can be performed in direct sequence spread spectrum (DSSS) or frequency hopping spread spectrum (FHSS). For voice traffic we reserve a channel, while for data traffic the access is random using a CSMA/CA mechanism, and in this case a selective reject and push-out mechanism is added to meet the quality of service required by data traffic. To study the performance of the proposed protocol and to show the benefits of our design, a mathematical model is built based on Markov chains. The system could be represented by a Markov chain which is difficult to solve as the state-space is too large. This is due to the resource management and user mobility. Thus, we propose to build an aggregated Markov chain with a smaller state-space that allows performance measures to be computed easily. We have used stochastic comparisons of Markov chains to prove that the proposed access protocol (with selective reject and push-out mechanisms) gives less loss rates of high priority connections (data and voices) than the traditional one (without admission policy and selective reject and push-out mechanisms). We give numerical results to confirm mathematical proofs.**

Index Terms: **MAC protocols, performance evaluation, quality of service, stochastic comparisons, wireless networks.**

I. INTRODUCTION

The development of the IEEE 802.11 protocol has been very important over the last few years. The reasons behind the success of this wireless technology are the high throughput (at a maximum of 54 Mbps in the IEEE 802.11a/g standard) without using wire and the ease of deployment (in a house, small company etc.). Contrary, this technology has some weaknesses from the quality of service (QoS) point of view. Under the random access using carrier sense multiple access with collision avoidance-distributed coordination function (CSMA/CA-DCF), the QoS is not guaranteed for some applications that require a certain level of QoS like multimedia applications.

In this study, we propose an improvement of the CSMA/CA protocol used in 802.11 in order to enhance the QoS for some applications. We focus on the distributed coordination function (DCF) period. The originality of the multi-channel CSMA/CA is to make the transmission of multiple user possible at the same moment, using different chipping code in direct sequence spread spectrum (DSSS), or frequency hopping in frequency hopping spread spectrum (FHSS) [1].

To show the benefits of the proposed protocol, we evaluate its performance. The system associated to the proposed protocol can be represented by a Markov chain with a large state-space. It is clear that the numerical resolution of this chain is complicated; this is due to the explosion of the state-space. Thus, we propose to carry out this evaluation using a methodology whose application is quite recent in the context of wireless networks: The stochastic comparison of Markov chains. This methodology is based on the theory of stochastic orderings (see [2]–[5]). Its application to the performance evaluation of computers and communication systems is widely used [6]. In [7] and [8], stochastic comparisons were used in order to compute loss rates bounds in an ATM node. In [9], it is used to compute the bounds on dropping handover in mobile networks. In [10], we make the performance evaluation of a MAC protocol in mobile networks. Experiments have proved for these applications that bounds provide interesting results for performance studies.

This paper is organized as follows. After a description of the CSMA/CA in a single and in a multi-channel environment in Section II, our new MAC protocol is presented in Section III. Section IV describes the mathematical model associated with our protocol. Next, in Section V, using stochastic comparisons we prove that the proposed protocol improves performance measures as dropping handover and call blocking. Some numerical results are presented in Section VI to confirm the mathematical proofs. Finally, the last section concludes the main contributions of this work and gives future prospects.

II. MAC PROTOCOL IN IEEE 802.11

A. Introduction

The Data link layer of standard 802.11 is composed of two sub-layers;

- the logical link control (LLC) layer
- medium access control (MAC) layer.

The 802.11 standard defines two forms of medium access, the DCF and the point coordination function (PCF). The mode of access in DCF is random, while in PCF is deterministic.

Manuscript received May 9, 2005; approved for publication by Dong Geun Jeong, Division II Editor, October 25, 2007.

J. Ben-Othman is with the Laboratoire Prism-CNRS, Université de Versailles, France, email: jalel.ben-othman@prism.uvsq.fr.

H. Castel is with the Dept. RST/SAMOVAR, Institut National des Télécommunications, France, email: hind.castel@int-evry.fr.

L. Mokdad is with Laboratoire Lamsade, Université de Paris Dauphine, France, email: lynda.mokdad@lamsade.dauphine.fr.

To support time-bounded delivery of data frames, the 802.11 standard defines the optional PCF where the access point grants to an individual station access to the medium by polling the station during the contention free period. Stations cannot transmit frames unless the access point polls them first. The period of time for PCF-based data traffic (if enabled) occurs alternately between contention (DCF) periods.

DCF is based on the CSMA/CA protocol which we will describe in details bellow.

B. CSMA/CA Description Protocol

B.1 Protocol Description

CSMA/CA is the basic medium access control method employed in IEEE 802.11 wireless networks. The IEEE 802.11 uses collision avoidance rather than collision detection which is used in wired systems such as Ethernet (IEEE 802.3). Unlike the station in a wired network, a WLAN's station cannot detect a collision while transmitting as it operates in half duplex and where we cannot assume that all stations can hear each other. Thus, unlike carrier sense multiple access/collision detect (CSMA/CD) which deals with transmissions after a collision has occurred, CSMA/CA acts to prevent collisions before they happen. If a collision occurs, the transmitting station will not receive an acknowledgement from the intended receiving station. For this reason, acknowledgement packets have a higher priority than all other network traffic.

Upon the completion of data transmission, the receiving station transmits an acknowledgement packet before any other node can begin transmitting a new data packet. A common problem in WLANs is the hidden node problem. This problem could be described as follows: assume that we have three nodes A, B, and C, that are arranged such that A and B are in mutual range, B and C are in mutual range, but A and C cannot hear each other. Assume that A starts to transmit a packet to B and some time later node C also decides to start a packet transmission. A carrier sensing operation by C shows an idle medium since C cannot hear as signals. When C starts its packet transmission, the signals collide at B and both packets are useless. The hidden nodes are solved by the use of an request-to-send (RTS)/clear-to-send (CTS) protocol prior to packet transmission. This solution is based on the multiple access collision avoidance (MACA) protocol. In a three nodes (A, B, and C) network, node A sends a small RTS packet which is heard by node B which sends a small CTS packet which is heard by both nodes A and C. Node C will not transmit in this case. The collision will be avoided. 802.11 was enriched also by the virtual carrier sense mechanism in order to reduce the probability of two nodes colliding because they cannot hear each other. A node wanting to transmit a packet will first transmit an RTS which will include the source, destination, and the duration of the following transaction. The destination node will respond (if the medium is free) with a CTS which will include the same duration information. All nodes receiving either the RTS and/or the CTS, will set their network allocation vectors (NAVs), for the given duration, and will use this information when sensing the medium (see Fig. 1). When an 802.11 station intends to transmit a message, it will first sense whether another station is already transmitting (carrier sense). If no other transmissions are sensed, the 802.11 device will send a small RTS packet to its intended receiver. If the receiver senses that the medium is clear, it sends a CTS packet in reply. Once the station wishing to transmit receives the CTS packet, it sends the actual data packet to its intended recipient. If the transmitting station does not receive a CTS packet in reply, it begins the RTS procedure over again. If a node does sense another transmission when it wants to send, it will apply a random deferral timer procedure called 'backoff.' The randomly generated backoff delay is uniformly chosen in the range $[0, W-1]$ where W is called the backoff window or contention window (CW). Every time the node which needs to send data senses a busy channel, it doubles the contention window (the range of values from which a random backoff time is chosen) and generates a new random backoff period to wait before the next attempt. This is known as exponential backoff and leads to an effect known as the channel capture effect, whereby a successfully transmitting node will reset its contention window and is able to send data, whereas other competing hosts double their contention window. This means that for all successive retransmissions, the value of CW increases exponentially (i.e., $CW_{new} = CW_{old} \times 2 - 1$), until it reaches and then stays at CW_{max}. CW will be reset to CW_{min} after a successful transmission. The backoff method is used to minimize collisions and maximize throughput for both low and high network utilizations.The result on channel efficiency is beneficial as a node that 'wins' a contest can transmit for longer, without having to waste bandwidth in other contests; but the effect on delay and delay variance is not so advantageous.

The 802.11 standard defines various frame types that nodes use for communications, as well as managing and controlling the wireless link. Every frame has a control field that depicts the 802.11 protocol version, frame type, and various indicators, such as whether WEP is used, power management is active, and so on. In addition, all frames contain MAC addresses of the source and destination stations (and access point), a frame sequence number, frame body and frame check sequence (for error detection). The MAC layer manages and maintains communications by coordinating the access to the shared radio channel and by using protocols that enhance communications over the wireless medium. Often when talking about the access to the channel, one limits this to the MAC layer, whereas it relates to all layers. We could improve the QOS hugely if we try to find out certain cooperation between the various layers.

Our protocol works in multichannel, meaning that it is possible to mix several communications on different channels. There are two different techniques of multiplexing, the DSSS and the FHSS. More details are given in the next section.

B.2 Multichannel CSMA/CA

The above access protocol on a single channel is prone to inefficiencies at heavy loads, since as the traffic increases, the bandwidth is wasted due to collisions and backoffs. Collisions can occur among the control packets (such as RTS and CTS). The un-synchronization in back-off delays yields an idle medium when all contending nodes are in backoff state, which highly degrades the over all throughput.

In addition, any node hearing RTS or CTS must defer at

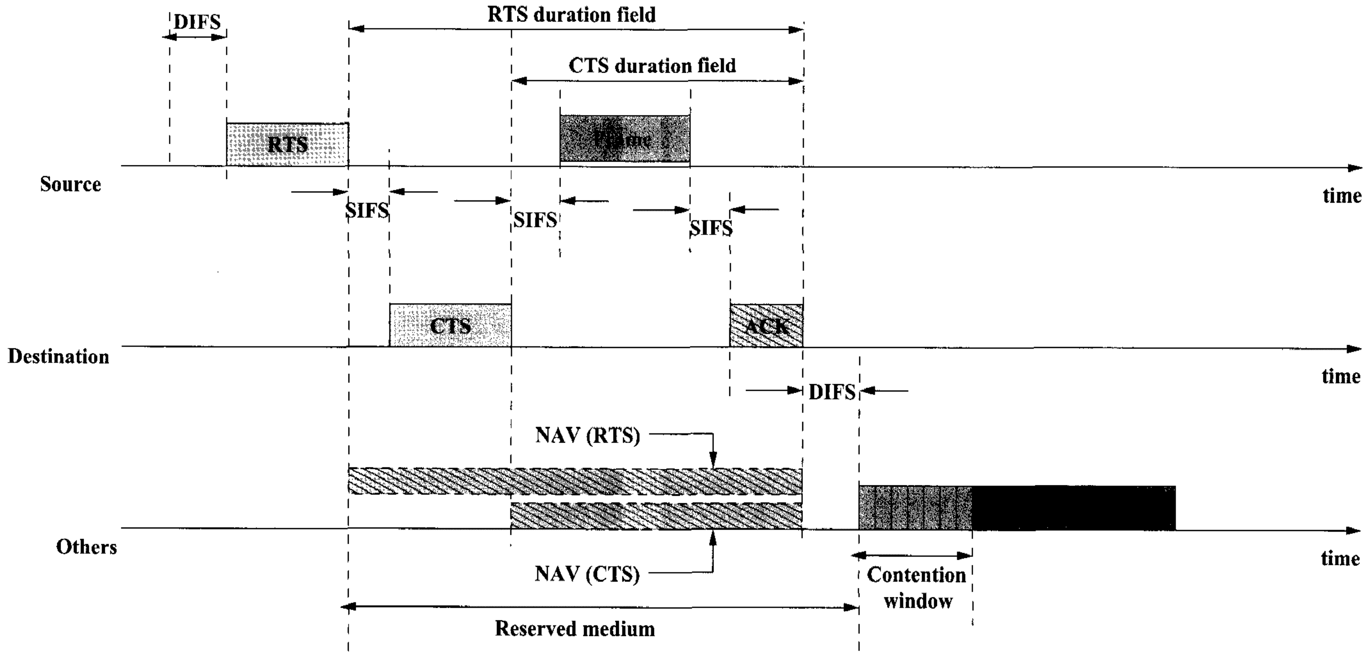

Fig. 1. RTS/CTS mechanism in 802.11.

least until the end of the entire exchange (i.e., until end of ACK). This means that concurrent transmissions cannot take place when two senders hear each other, even though the respective receivers do not hear any node other than their respective senders (the so called "exposed terminal problem"). This is because each sender needs to be able to receive control packets (CTS and ACK) correctly. The use of multiple channels provides some performance advantages in reducing collisions and enabling more concurrent transmissions and hence better bandwidth. Multichannel protocols allow a number of nodes in the same neighborhood to transmit concurrently on different channels without interfering with one another. Carrier sensing can be coupled with an efficient channel selection mechanism to pick the clearest channel for transmission. If multiple channels are formed on the basis of multiple CDMA codes, a receiver may also receive multiple signals from different sources at the same time. This approach was applied to both DSSS and FHSS systems [11]. In the DSSS method, the stream of information to be transmitted is divided into small pieces, each of which is allocated to a frequency channel in the spectrum. A data signal at the point of transmission is combined with a higher data-rate bit sequence (also known as a chipping code) that divides the data according to a spreading ratio. The redundant chipping code helps the signal resist interference and also enables the original data to be recovered if data bits are damaged during transmission. In FHSS a broad slice of the bandwidth spectrum is divided into many possible broadcast frequencies. The stream of information is transmitted on different frequencies [12]. The weakness of these methods is that bandwidth allocation between several mobile nodes is not made according to the quality of service of the source. Thus, we have developed a new MAC protocol in multichannel system that ensures the QoS of the source. We will describe our method in the next section.

III. A MULTI-SERVICE MAC PROTOCOL FOR IEEE 802.11 NETWORKS IN DCF MODE

A. Introduction

The 802.11 Wireless LANs are growing quickly and the services offered by this type of networks are increasingly varied. In 802.11 networks, various applications and data types are supported, even voice or video. This requires the development of QoS techniques to provide effective resource management and fast access to the network. In DCF mode the 802.11b or 802.11g protocol uses CSMA/CA access. This method is efficient at low loads of the frequency coverage (cell). However, at a high loads this method is not efficient specially for sources that need quality of service, such as voice or multimedia.

In this study, to improve quality of service of MAC protocol, we consider users with different QoS requirements. The services considered are voice and data, the latter includes services such as multimedia. To differentiate between services, we consider three priority levels for voice connections:

- High: For emergency calls,
- Medium: For users with a high access priority to the network,
- Low: For users without priority.

For voice we distinguish the new calls, and handovers which correspond to a call coming from a neighboring cell. We also assign two levels of priority for data services with high and low priority, where high priority level corresponds to a service with QoS requirement like multimedia and low priority to a best-effort service.

B. Allocation Policy Description on Air Interface

To ensure quality of service for voice connections in a CSMA/CA multichannel we assign a reserved channel. Thus, in DSSS it means that we assign a chipping code for a voice connection and this code is only reserved for one connection. In FHSS, we assign a frequency hopping for one voice call. For compatibility reasons with existing systems (e.g., IEEE 802.11b/g), the DSSS method is considered.

Initially data traffic is stored in a buffer if there is space available; otherwise the selective reject and push-out mechanism is used to serve data at high priority. On the air interface we use a CSMA/CA method on the available channel.

We also add a priority policy between voice and data: We assume is N the capacity of the access point (it corresponds to the number of different existing chipping codes in DSSS), we consider two thresholds (N_1 and N_2). For voice traffic, under N_1, we accept all handovers and new connections without distinguishing priorities. In the second range (between N_1 and N_2), we accept handovers and new connections with both high and medium priorities. In the last range (between N_2 and the channel capacity N), we accept only handovers and connections with high priority. For data services we use the push-out mechanism if the buffer is full, in order to serve data packets with high priority. The thresholds are computed in order to have dropping handover less than 10%. The performance evaluation of the protocol helps us to find the best thresholds. In the following section, a mathematical model for the proposed protocol is described, which is used later for performance evaluation.

IV. PROTOCOL MODELIZATION

The multichannel is represented by N channels. In DSSS, N represents the number of the chipping code, in FHSS, N represents the number of frequency hopping. Each channel can be occupied by data or voice communications. A buffer with limited capacity, B, is used to store data packets. If a data packet arrives and all channels are occupied, the data packet is stored in the buffer. But if a voice connection arrives and all channels are busy, the communication is lost.

Assumptions for call arrivals and service processes are made according to studies in [13]: The voice connections are generated according to an independent Poisson process with rates, λ_{Hv} for the high priority, λ_{Mv} for the medium priority, and λ_{Lv} for the low priority. The distribution of the communication time for voice is assumed to be an exponential distribution with rate μ_v. The arrivals of handoff voice from other cells to this cell is assumed to be a Poisson process with rates, γ_{Hv} for the high priority, γ_{Mv} for the medium priority and γ_{Lv} for the low priority. We also assumed that the arrivals of data packets follow a Poisson process with rates λ_{Hd} for high priority and λ_{Ld} for low priority. The distribution of the communication time for a data packet is assumed to be an exponential distribution with rate μ_d. The proposed system can be represented by a continuous time Markov chain $\{X^p(t), t \geq 0\}$. The state of this Markov chain at time t can be given by the vector:

$$X^p(t) = (N_s^p(t), N_b^p(t), N_{b_1}^p(t), Tab(t))$$

where at any time t:

- $N_s^p(t)$ is the number of channels occupied by voice connections and data packets.
- $N_b^p(t)$ is the number of data packets in the buffer.
- $N_{b_1}^p(t)$ is the number of high priority data packets in the buffer.
- $Tab(t)$ represents the buffer state. A list of 0s and 1s indicate the priority level of the data cell waiting for a service; 0 for medium priority, and 1 for high priority.

Now, we will explain how voice connections are accepted using their priorities. We apply the selective reject and push-out mechanisms as follows: we accept a connection according to its priority and the number of busy channels. We consider two values N_1 and N_2 such that: $N_1 < N_2 < N$. These values allow us to define conditions to accept voice connections using their priorities. We have four cases for a state $X^p(t)$ as explained below:

- If $N_s^p(t) < N_1$ then all voice connections are accepted, so the proposed system acts like a classical CSMA/CA.
- If $N_1 \leq N_s^p(t) < N_2$ all voice connections are accepted except those with low priorities are refused.
- If $N_2 \leq N_s^p(t) < N$ only high priority voice connections are accepted, while those with medium and low priorities are refused.
- If $N_s^p(t) = N$ all voice connections are refused.

For the data packets, we choose a buffer policy which uses the push-out mechanism [14]:

- When the buffer is not full, low and high priority packets can be stored in the buffer.
- When the buffer is full, an arriving low priority packet is lost, while an arriving high priority packet pushes out of the buffer a low priority packet, if there are any in the buffer, otherwise the high priority packet is lost. The deletion discipline is LIFO.

As we have explained how the process performs, below we derive its evolution equations.

A. Evolution Equations

In this section, we give the evolution equations of the proposed system. We must remember that we are in a continuous time, so we can have only one event at any time t. Thus, in each evolution equation, we can have only either an arrival or a service.

Let us define the number of arrivals (0 or 1) in the time interval $[t, t+dt]$ of voices (new connections and handover): $A_{Hv}(t+dt)$ for high priority, $A_{Mv}(t+dt)$: for medium priority, and $A_{Lv}(t+dt)$: for low priority. And similarly for data packets $A_{Hd}(t+dt)$ for high priority, and $A_{Ld}(t+dt)$ for low priority. We complete this notation by: $A_v(t+dt) = A_{Hv}(t+dt) + A_{Mv}(t+dt) + A_{Lv}(t+dt)$, the total number of arrivals of voice connections, and by $A_d(t+dt) = A_{Hd}(t+dt) + A_{Ld}(t+dt)$ the total number of data packets arrivals. As the service discipline is FIFO, in the buffer, we denote by head(t) the first data packet waiting in the buffer at time t. Let us explain now the evolution of the process. We will explain how each component of the vector $X^p(t)$ changes. We begin with the first component $N_s^p(t)$:

$$\begin{aligned} N_s^p(t+dt) = {} & N_s^p(t) \\ & + A_v(t+dt) * 1_{N_s^p(t)<N_1} \quad (1) \\ & + A_{Hv}(t+dt) * 1_{N_1 \leq N_s^p(t)<N_2} \quad (2) \end{aligned}$$

$$+A_{Mv}(t+dt) * 1_{N_1 \le N_s^p(t) < N_2} \quad (3)$$
$$+A_{Hv}(t+dt) * 1_{N_2 \le N_s^p(t) < N} \quad (4)$$
$$+A_d(t+dt) * 1_{N_s^p(t) < N} \quad (5)$$
$$+1_{N_b^p(t)>0} * 1_{N_s^p(t)<N} \quad (6)$$
$$-1_{N_s^p(t)>0}. \quad (7)$$

$N_s^p(t)$ increases if there is an arrival of a voice connection and according to its priority (lines 1, 2, 3, 4), and as well if there is an arrival of a data (line 5). If there is a service and the buffer is not empty, then the data packet transits from the buffer to the server, so the component does not change (lines 6 and 7 are executed). $N_s^p(t)$ decreases if there is a service (of voice or data) (line 7), and the buffer is empty. For $N_b^p(t)$, we have the following evolution equation:

$$N_b^p(t+dt) = N_b^p(t)$$
$$+A_d(t+dt) * 1_{N_s^p(t)=N} * 1_{N_b^p(t)<B} \quad (8)$$
$$-1_{N_s^p(t)<N} * 1_{N_b^p(t)>0}. \quad (9)$$

$N_b^p(t)$ increases if there is an arrival of data and all the servers are busy (line 8), and decreases if a server becomes free (line 9). The evolution equation of $N_{b_1}^p(t)$ is:

$$N_{b_1}^p(t+dt) = N_{b_1}^p(t)$$
$$+A_{Hd}(t+dt) * 1_{N_s^p(t)=N} * 1_{N_{b_1}^p(t)<B} \quad (10)$$
$$-1_{head(t)=1} * 1_{N_s^p(t)<N}. \quad (11)$$

$N_{b_1}^p(t)$ increases if there is an arrival of a high priority data packet in the buffer (line 10), and decreases if the first data packet waiting in the buffer has a high priority (line 11). The evolution equation of Tab is:

$$Tab(t+dt) = Tab(t)$$
$$-head(t) \quad (12)$$
$$+A_d(t+dt). \quad (13)$$

The evolution of the vector Tab is represented by a data packet arrival at the end of the vector, and by a service at the head of this vector according to the FIFO discipline.

V. PERFORMANCE EVALUATION

In this section, we prove that the proposed protocol provides less dropping handover and less call blocking of high priority connections than a classical CSMA/CA multi-channel MAC protocol. We compare stochastically the Markov chains of both systems. We compute the dropping handover and the call blocking of high priority data packets and high priority voices. The results obtained confirm the proof and show that the proposed system significantly improves the loss rates. The comparison of these performance measures is based on the stochastic ordering theory. Some definitions and theorems are given in the appendix.

In the classical CSMA/CA multi-channel MAC protocol, as we don't accept connections according to their priority, we only need to know the number of data packets waiting in the buffer, and the number of busy channels. Thus, states of this model can be represented at time t by:

$$X^c(t) = (N_s^c(t), N_b^c(t))$$

where:

- $N_s^c(t)$ is the number of equipped channels by voice connections and data packets at time t.
- $N_b^c(t)$ is the number of data packets in the buffer at time t.

Now, we give the evolution equations of the system: First, we give $N_s^c(t)$:

$$N_s^c(t+dt) = N_s^c(t)$$
$$+(A_v(t+dt) + A_d(t+dt)) * 1_{N_s^c(t)<N} \quad (14)$$
$$+1_{N_b^c(t)>0} * 1_{N_s^c(t)<N} \quad (15)$$
$$-1_{N_s^c(t)>0}. \quad (16)$$

The component $N_s^c(t)$ increases if there is an arrival of a voice or a data packet (line 14). If there is a service and the buffer is not empty, then the data packet transits from the buffer to the server, so the component does not change (lines 15 and 16 are executed). The component decreases when a service finishes and the buffer is empty (line 16). Let give $N_b^c(t)$:

$$N_b^c(t+dt) = N_b^c(t)$$
$$+A_d(t+dt) * 1_{N_s^c(t)=N} * 1_{N_b^c(t)<B} \quad (17)$$
$$-1_{N_b^c(t)>0} * 1_{N_s^c(t)<N}. \quad (18)$$

The component $N_b^c(t)$ increases if there is an arrival of a data packet and all the servers are busy (line 17), and decreases if a server becomes free (line 18). In the view of evolution equations and assumptions made on arrivals and services, we can deduce that $N^c(t)$ is a continuous-time Markov chain.

A. Proofs

We will prove that loss rates of high priority data packets, dropping handover, and call blocking of high priority voices are lower using our defined protocol. As the systems are not defined on the same state-space, let S be the many to one mapping which maps the state space of $X^p(t)$ into the state space denoted ϵ of $X^c(t)$ as follows: $S(X^p(t)) = (N_s^p(t), N_b^p(t))$.

As we compare the systems on ϵ, we define on it the known order (component-by-component order) denoted by $\preceq$.

Let $x = (x_1, x_2) \in \epsilon$ (resp. $y = (y_1, y_2) \in \epsilon$), we say that: $x \preceq y \Leftrightarrow x_1 \le x_2 \text{ and } y_1 \le y_2$. From the order $\preceq$ defined on ϵ, we compare stochastically the processes, and we give the following theorem:

Theorem 1: $\{S(X^p(t)), t \ge 0\}) \preceq_{st} \{X^c(t), t \ge 0\}$.

Now, we prove this theorem using the fundamental Theorem 2 (see the appendix) of the coupling of the processes.

To do this, we suppose that:

$$S(X^p(t)) \preceq X^c(t)$$

and we prove that:

$$S(X^p(t+dt)) \preceq X^c(t+dt).$$

We will compare evolutions of the systems considering that the same event happens in both systems.

Next, we give a detailed step by step proof of the Theorem 1.

A.1 Coupling of the Processes

In such systems, we can have an arrival of a voice or a data packet and a service of a voice or data packet. We develop the two cases of events below:

■ **Arrival Cases:**

An arrival of a voice: We can see in the following equations that, for the classical CSMA/CA, the number of the busy servers increases for all kinds of arrival voices but for the proposed system, it depends on the number of busy servers and the priority of voices.

► **If at least one server is free** corresponding to the condition $N_s^c(t) < N$, then $X^c(t+dt) = (N_s^c(t)+1, N_b^c(t))$, and according to the voice priority in the proposed system, then we have:

- If $(N_s^p(t) < N_1)$ or $(N_1 \leq N_s^p(t) < N_2)$ and the arrival voice connection is with high or a medium priority) or $(N_2 \leq N_s^p(t) < N)$ and the arrival voice connection is with high priority, then $S(X^p(t+dt)) = (N_s^p(t)+1, N_b^p(t))$.
- If $(N_1 \leq N_s^p(t) < N_2)$ and the arrival voice connection is with low priority or $(N_2 \leq N_s^p(t) < N)$ and the arrival voice connection is with medium or low priority, then $S(X^p(t+dt)) = (N_s^p(t), N_b^p(t))$.

► **If all servers are busy** corresponding to the condition $N_s^c(t) = N$ then $X^c(t+dt) = (N, N_b^c(t))$ and for the proposed system we have two cases:

- $N_s^p(t) < N$, then $S(X^p(t+dt)) = (N_s^p(t)+1, N_b^p(t))$ or $S(X^p(t+dt)) = (N_s^p(t), N_b^p(t))$, according to the voice priority, we can find the details in the precedent case.
- $N_s^p(t) = N$, then $S(X^p(t+dt)) = (N, N_b^p(t))$.

So, **we have proved that for a voice arrival, if** $S(X^p(t)) \preceq X^c(t)$**, then** $S(X^p(t+dt)) \preceq X^c(t+dt)$**.**

An arrival of a data packet: If there is a free server in both systems, the data packet is served; otherwise, if all servers are busy and the buffer is not full in both systems, then the number of data packets waiting in the buffer increases by one. In the classical CSMA/CA, if the buffer is full, then the data packet is lost. However in our proposed system, according to the push-out mechanism, it depends on the data packet priority. If it is a low priority data packet it is lost, but if it is a high priority data packet, it replaces a low priority data packet in the buffer if there are any, otherwise it is lost. So in this case, the number of data packets waiting in the buffer is still equal to B in both systems. So clearly, these are all the possible cases:

► **If all servers are busy and the buffer is not full** corresponding to the condition $N_s^c(t) = N$ and $N_b^c(t) < B$ then $X^c(t+dt) = (N, N_b^c(t)+1)$, and for the proposed system we have two cases:

- If $N_s^p(t) = N$, then we have the same evolution (i.e., $S(X^p(t+dt)) = (N, N_b^p(t)+1)$).
- If $N_s^p(t) < N$, then we have $S(X^p(t+dt)) = (N_s^p(t)+1, N_b^p(t))$.

► **If all servers are busy and the buffer is full** corresponding to the condition $N_s^c(t) = N$ and $N_b^c(t) = B$ then we have $X^c(t+dt) = (N, B)$, and for our proposed system, we have these cases:

- If $N_s^p(t) = N$ and $N_b^p(t) = B$ then we have the same evolution (i.e. $S(X^p(t+dt) = (N, B))$.
- If $N_s^p(t) = N$ and $N_b^p(t) < B$ then we have $S(X^p(t+dt)) = (N, N_b^p(t)+1)$.
- If $N_s^p(t) < N$ and $\forall N_b^p(t)$ we have $S(X^p(t+dt)) = (N_s^p(t)+1, N_b^p(t))$.

► **If at least one server is free**, corresponding to the condition $N_s^c(t) < N$ then for the classical system, we have $X^c(t+dt) = (N_s^c(t)+1, N_b^c(t))$, and for the proposed system, we have the same state of evolution $S(X^p(t+dt)) = (N_s^p(t)+1, N_b^p(t))$.

So **we have proved that if:** $S(X^p(t)) \preceq X^c(t)$**, it is clear that** $S(X^p(t+dt)) \preceq X^c(t+dt)$.

■ **Service cases:**

Service happens in the same way in both systems, so the order is kept between the processes.

► **If the buffer is empty**, corresponding to the condition $N_b^p(t) = 0$, then a server can become free: $S(X^p(t+dt)) = (N_s^p(t)-1, 0)$, and for the proposed system, we have two cases:

- If $N_b^c(t) = 0$ then we have the same evolution, i.e., $X^c(t+dt) = (N_s^c(t)-1, 0)$.
- If $N_b^c(t) > 0$ then we have $X^c(t+dt) = (N_s^c(t), N_b^c(t)-1)$, i.e., one packet from a buffer takes the server which becomes free after a packet service.

► If $N_b^p(t) > 0$, then $S(X^p(t+dt)) = (N_s^p(t), N_b^p(t)-1)$, and $X^c(t+dt) = (N_s^c(t), N_b^c(t)-1)$.

We have proved that if we have the condition: $S(X^p(t)) \preceq X^c(t)$ then for every event that happens, we have: $S(X^p(t+dt)) \preceq X^c(t+dt)$. Thus, if we have the initial condition: $S(X^p(0)) \preceq X^c(0)$ then $S(X^p(t)) \preceq X^c(t), \forall t > 0$, so Theorem 1 is proved.

The stochastic comparison of Markov processes induces the stochastic comparison of their steady state distributions. Let Π^c (resp. Π^p) be the steady state distribution of the classical system (resp. the proposed system). We denote by $\Pi^c(x)$ (resp. $\Pi^p(y)$) be the stationary probability on state $x = (x_s, x_b)$ (resp. $y = (y_s, y_b, y_{b_1}, tab)$). We have this inequality:

$$\sum_{y|S(y)\in\Gamma} \Pi^p(y) \leq \sum_{x|x\in\Gamma} \Pi^c(x), \ \forall\Gamma \in \Phi_{st}(\epsilon). \quad (19)$$

A.2 Performance Measures Comparisons

We will explain how we compute the loss rates of high priority data, dropping handover and call blocking of high priority voices. Let R_{Hd}^p (resp. R_{Hd}^c) be the loss rate of high priority data packets, R_{Hv}^p (resp. R_{Hv}^c) be the dropping handover of high priority voice connections and R_{Bv}^p (resp. R_{Bv}^c) be the call blocking of high priority voice connections for the proposed system (resp. classical system). For the classical model, we have loss of high priority data if the buffer is full and all the servers are busy. The loss rate of high priority data packets for this system is:

$$R_{Hd}^c = \Pi^c(N, B) * \lambda_{Hd}.$$

For our proposed system, we have loss of high priority data packets only if the buffer is full of high priority packets, and when all servers are busy

$$R_{Hd}^{p} = \sum_{\substack{y|S(y)=(N,B)\\ y_{b_1}=B}} \Pi^p(y) * \lambda_{Hd}.$$

Let us now give the loss rate of high priority voices. In both systems, high priority voices are lost if all the servers are busy. The dropping handover for the proposed system is:

$$R_{Hv}^{p} = \sum_{y|y_s=N} \Pi^p(y)\gamma_{Hv}$$

and for the classical system is:

$$R_{Hv}^{c} = \sum_{x|x_s=N} \Pi^c(x)\gamma_{Hv}$$

and the call blocking for the proposed system is:

$$R_{Bv}^{p} = \sum_{y|y_s=N} \Pi^p(y)\lambda_{Hv}$$

and for the classical system is:

$$R_{Bv}^{c} = \sum_{x|x_s=N} \Pi^c(x)\lambda_{Hv}.$$

Now we will show that there are inequalities between the loss rates given above. We have these relations:

Proposition 1: $R_{Hd}^{p} \leq R_{Hd}^{c}$, $R_{Hv}^{p} \leq R_{Hv}^{c}$, and $R_{Bv}^{p} \leq R_{Bv}^{c}$.

Now, we will prove these inequalities. We begin with the first one. We use equation 19. This inequality is verified for all increasing sets of $\phi_{st}(\epsilon)$. So, for the particular increasing set $\Gamma = \{(N,B)\} \uparrow = \{(N,B)\}$ it is also verified. So we obtain this inequality:

$$\sum_{y|S(y)=(N,B)} \Pi^p(y) \leq \Pi^c(N,B).$$

It is clear that:

$$\sum_{y|S(y)=(N,B)} \Pi^p(y) = \sum_{\substack{y|S(y)=(N,B)\\ y_{b_1}=B}} \Pi^p(y) + \sum_{\substack{y|S(y)=(N,B)\\ y_{b_1}=B}} \Pi^p(y). \qquad (20)$$

In the proposed system, the previous equation means that when the buffer is full, we can deduce two cases, one when the buffer is full of only high priority data packets or different kinds of data packets. We deduce from the two previous inequalities the following:

$$\sum_{\substack{y|S(y)=(N,B)\\ y_{b_1}=B}} \Pi^p(y) \leq \sum_{y|S(y)=(N,B)} \Pi^p(y) \leq \Pi^c(N,B).$$

So, we can establish this inequality:

$$\sum_{\substack{y|S(y)=(N,B)\\ y_{b_1}=B}} \Pi^p(y) \leq \Pi^c(N,B).$$

If we multiply both components of the previous inequality by λ_{Hd}, we obtain the comparison of the recompense function: $R_{Hd}^{p} \leq R_{Hd}^{c}$. Now, let us compare the loss rates of high priority voice connections. In (19), we take the increasing set $\Gamma = \{(N,0)\} \uparrow = \{(N,0),(N,1),(N,2)....(N,B)\}$ then we obtain the following inequality:

$$\sum_{y|y_s=N} \Pi^p(y) \leq \sum_{x|x_s=N} \Pi^c(x).$$

So we obtain the comparison of the dropping handover of high priority voice connections:

$$R_{Hv}^{p} \leq R_{Hv}^{c}.$$

And the comparison of the call blocking of high priority voices:

$$R_{Bv}^{p} \leq R_{Bv}^{c}.$$

B. Aggregated Markov Process

As the numerical computation of the steady state of $X^p(t)$ is hard to manage due to the explosion of the state-space of the Markov chain, therefore an aggregated Markov process is proposed. Since loss rates can be computed using only two components: The number of busy servers and the number of data packets waiting in the buffer, we defined an aggregated Markov process called $X^{ag}(t)$ on state-space ϵ, which represents a lower bound of $S(X^p(t))$. The transition rates of the aggregated process $X^{ag}(t)$ are defined in order to provide lower bounds for performance measures. The computation of these transition rates are based on the works given in [2] and [5]. Now, we give an idea of computing the transition rates of the aggregated process. Transition rates of the aggregated Markov process $X^{ag}(t)$ are defined using the coupling theorem (Theorem 2 in the appendix), in order to have the coupled process $(X^{ag}(t), S(X^p(t))$ staying in:

$$K = \{(x,y) \in \epsilon \times E \mid x \leq S(y)\}.$$

The key idea is to have:

- If $X^{ag}(t)$ increases then $S(X^p(t))$ must increase also.
- If $S(X^p(t))$ decreases then $X^{ag}(t)$ must decrease also.

And the transition rates of $X^{ag}(t)$ must verify these conditions.

We suppose that $x = (x_1, x_2)$, and $S(y) = (s_1, s_2)$. As transitions of the processes $S(X^p(t))$ and $X^{ag}(t)$ can generate an increasing (arrival) or a decreasing (service) of a state, then we have two cases:

1. Arrival case: If x increases which means that $x- > x^+$ (x_1 or x_2 increases), such that $x^+ \succ S(y)$, then we must have $y- > y^+$ ($S(y^+)$ has $s_1 = s_1 + 1$ or $s_2 = s_2 + 1$) such that $(x^+, S(y^+)) \in K$.
2. Service case: If y decreases which means that $y- > y^-$ ($S(y^-)$ (has $s_1 = s_1 - 1$ or $s_2 = s_2 - 1$), such that $x \succ S(y^-)$, then we must have $x- > x^-$ (x_1 or x_2 decreases) such that $(x^-, S(y^-)) \in K$.

Let Q^{ag} (resp. Q^p) transition rates matrices of $X^{ag}(t)$ (resp. $X^p(t)$).

Table 1. Parameter values.

Parameter	value
Number of Channels	20, 30
Capacity	11 Mbps
Throughput	9.6 kbps per user
Load	$0.2, \cdots, 0.9$
N1	40%
N2	70%

According to the conditions given previously, we obtain:

1. If the number of the busy servers is less than N_1 corresponding to the condition $x_1 < N_1$, $Q^{ag}(x, x^+) \leq Q^p(y, y^+)$ where $x^+ = (x_1 + 1, x_2) \preceq S(y^+) = (s_1 + 1, s_2)$ so $Q^{ag}(x, x^+) = \lambda_{Hv} + \lambda_{Mv} + \lambda_{Lv} + \lambda_{Hd} + \lambda_{Ld} + \gamma_{Hv} + \gamma_{Mv} + \gamma_{Lv}$.
2. For the other cases, increasing rates of the aggregated process are: if $N_1 \leq x_1 < N_2$, then $Q^{ag}(x, x^+) = \lambda_{Hv} + \lambda_{Mv} + \lambda_{Hd} + \lambda_{Ld} + \gamma_{Hv} + \gamma_{Mv}$, and if $N_2 \leq x_1 < N$, $Q^{ag}(x, x^+) = \lambda_{Hv} + \lambda_{Hd} + \lambda_{Ld} + \gamma_{Hv}$. If $x_1 = N$, then $Q^{ag}(x, x^+) = \lambda_{Hd} + \lambda_{Ld}$.
3. For the decreasing rates, the condition is: $Q^{ag}(x, x^-) \geq Q^p(y, y^-)$, where $(x_1 - 1, x_2) \preceq S(y^-) = (s_1 - 1, s_2)$. as $Q^p(y, y^-) = \mu_v$ or $= \mu_d$, then $Q^{ag}(x, x^-) = \max(\mu_v, \mu_d)$. Similarly with the same way, we obtain that: $Q^{ag}(x, x^-) = \max(\mu_v, \mu_d)$.

The transition rate matrix $Q^{ag}(t)$ is defined so as the Markov process $X^{ag}(t)$ is irreducible, then the stationary probability distribution exists, and performance measures bounds of loss rates and dropping handover can be computed.

VI. NUMERICAL COMPUTATIONS

To perform performance evaluation of the proposed protocol, we carry-out a set of experiments by varying some parameters like the number of channels (N) and the system load, and compare the results with the classical multi-channel CSMA/CA protocol. Experimental parameters are listed in Table 1.

In Fig. 2, we can see the dropping handover of high priority voices for both protocols versus the load. For this experiment, we have taken $N = 20$ and for data we have considered a throughput equal to 9600 b/s. In order to show the efficiency of the proposed protocol, a system with a high load of voices is considered. Thus, we have taken 15 seconds as the mean inter-arrival time of voice calls.

In Fig. 3, we can see the dropping handover of high priority voices for the both protocols versus the load. For this experiment, we have increased the channel number by taking $N = 30$.

We can see clearly in Figs. 2 and 3 that the proposed protocol gives lower values for dropping handover than the classical protocol. We notice in Fig. 4 that when we increase the number of channels, the dropping handover of high priority voices in proposed protocol decreases. In Figs. 5 and 6, we can see the packet loss probabilities of high priority data for both protocols versus the load. For these experiments, we have taken a mean inter-arrival time of voice call equal to 15 seconds, a data throughput equal to 9600 b/s and $N = 20$ in Fig. 5 and $N = 30$ in Fig. 6.

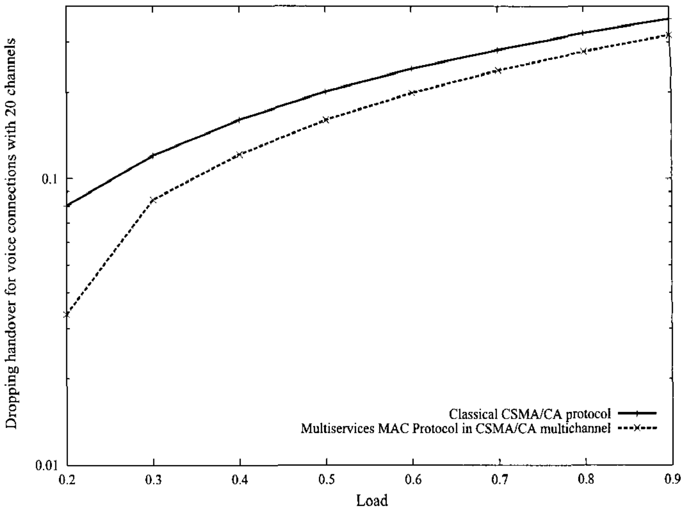

Fig. 2. Dropping handover for 20 channels.

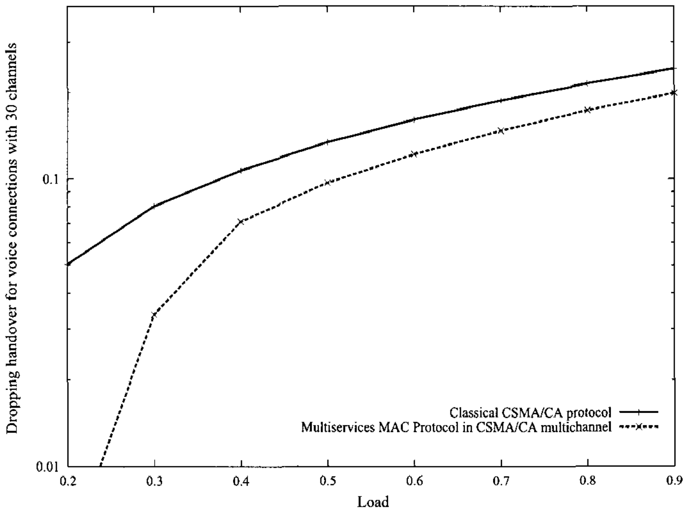

Fig. 3. Dropping handover for 30 channels.

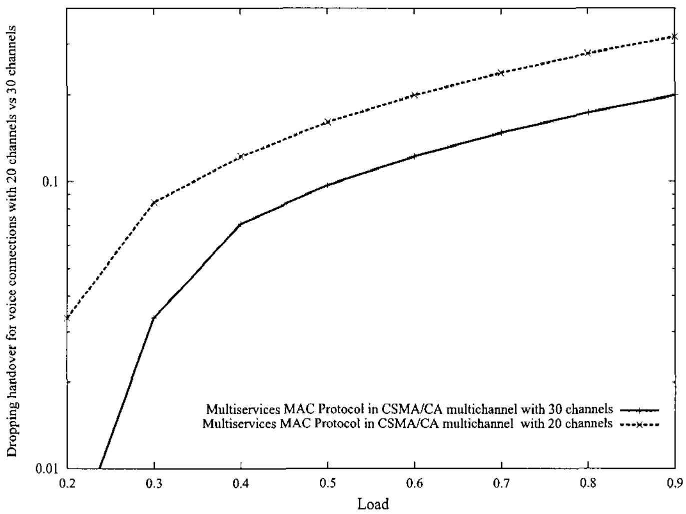

Fig. 4. Dropping handover with 20 and 30 channels.

We can observe in Figs. 5 and 6 that packet loss probabilities are lower in the proposed protocol than in the classical multichannel CSMA/CA. According to the previous results, we can conclude that the proposed protocol improves the performance measures for high priority communications.

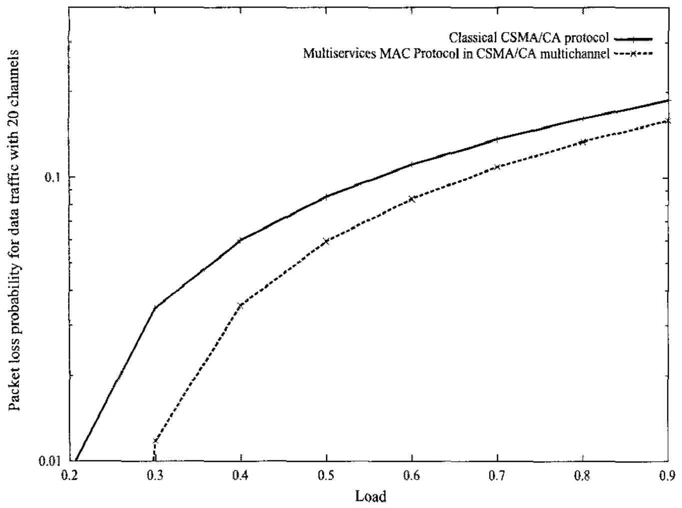

Fig. 5. Packet loss probabilities for 20 channels.

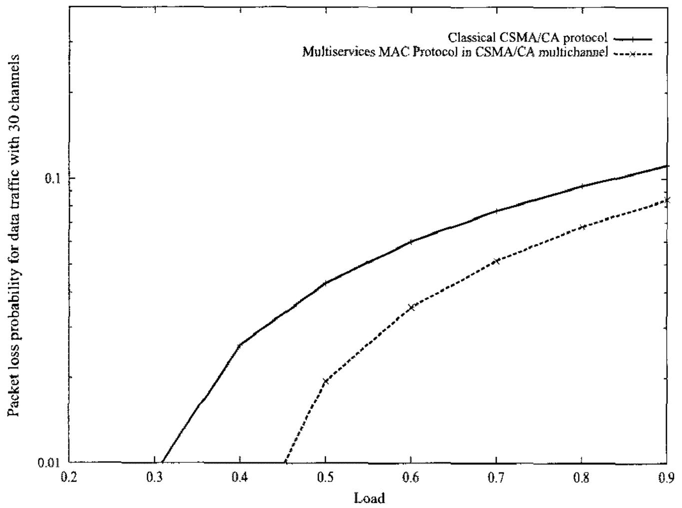

Fig. 6. Packet loss probabilities for 30 channels.

VII. CONCLUSION

In this paper, we have proposed and evaluated a new MAC protocol which integrates the quality of service for voice and data in IEEE 802.11 networks using a CSMA/CA multichannel. In order to ensure QoS for voice traffic, we reserved a channel for one voice call only. We have developed a new call admission scheme in order to ensure Quality of Service for a specific connection by the differentiation of three levels of call (high, medium and low priority). For data traffic, we have also defined two priority levels (high and medium), initially all data are stored in a buffer before transmission on air interface. In the case of the buffer is full, a selective reject and push out mechanisms for data with high priority is added. The data transmission occurs only on the available channel, which is not reserved for voice calls.

Our protocol is compared to the classical multi-channel CSMA/CA protocol. We have used stochastic comparisons of Markov chains in order to prove mathematically that the proposed protocol provides better performance measures than the traditional one. This method overcomes the disadvantages of the simulation and the numerical methods. In fact, simulation takes a long time and provides only approximate results. On the other hand, numerical methods give exact results but are very difficult to apply due to the explosion of the Markov chain state-space size. Thus, we have built an aggregated Markov chain with a less state-space size that allows easily performance measures to be computed using the numerical method [15]. We have computed numerically the dropping handover and call blocking of high priority communications for both protocols. Experiments confirm the mathematical proofs, in other words they show that the proposed protocol improves the QoS of high priority communications.

In this study, we focus only on the blocking probabilities, dropping handover or data loss rates, as parameters of quality of service. It is also important to take into account other parameters such as delay, jitter, and throughput. However, these parameters are not considered in our mathematical model. Thus, we plan to study them in future works.

APPENDIX

In this section, we give some definitions and theorems about stochastic orderings. We refer to Stoyan [3], Massey [4], and Lindvall [16] for a survey of stochastic bounding techniques applied in queuing theory. A stochastic ordering can be defined using two formalisms: increasing set or increasing functions. Let ε be a discrete denumerable state-space, and $\preceq$ be at least a pre order on ε. First, let us give the definition about an increasing set [4]. Let $\Gamma \subseteq \varepsilon$, we denote by

$$\Gamma \uparrow = \{y\epsilon\varepsilon | y \succeq x, \text{for } x\epsilon\Gamma\}.$$

Definition 1: *$\Gamma \subseteq \varepsilon$ is called an increasing set if and only if $\Gamma = \Gamma\uparrow$.*

Let $\preceq_\Phi$ be a stochastic ordering defined on ε. It is defined using the increasing set formalism if it can be generated by the increasing set family $\Phi(\varepsilon)$ as follows [4]:

Definition 2: *Let X (resp. Y), be a random variable on ε, with P (resp. Q) its probability measure on ε*

$$X \preceq_\Phi Y \Leftrightarrow P \preceq_\Phi Q \Leftrightarrow P(\Gamma) \leq Q(\Gamma), \forall\Gamma\epsilon\Phi(\varepsilon)$$

where $P(\Gamma) = \sum_{x\epsilon\Gamma} P(x)$.

Massey [4] has proved that $\Phi(\varepsilon)$ induces a stochastic ordering if and only if it is a strongly separating family of increasing sets. Let $\Phi_{st}(\varepsilon)$ and $\Phi_{wk}(\varepsilon)$ be two families of increasing sets which induce stochastic orderings: $\preceq_{st}$, $\preceq_{wk}$. These families are defined as follows:

$$\Phi_{wk}(\varepsilon) = \{\{x\}\uparrow, x\epsilon\varepsilon\} \cup \varepsilon$$

where $\{x\}\uparrow = \{y\epsilon\varepsilon | y \succeq x\}$ and $\Phi_{st}(\varepsilon) = \{\text{all increasing sets on } \varepsilon\}$.

$\preceq_{st}$ is the strong stochastic ordering, it is equivalent to the sample path ordering (see Strassen's theorem [3]), whereas $\preceq_{wk}$ is a weak ordering corresponding to tail distribution comparison. If the strong stochastic ordering exists between probability measures, then the weak ordering exists [4]:

Now, we will define the stochastic comparison of Markov processes [2], [4]. Let $\{X(t), t \geq 0\}$ (resp. $\{Y(t), t \geq 0\}$) defined on E (resp. F), and f a many to one mapping from E to ε (resp. g a many to one mapping from F to ε). We will compare stochastically the image $f(X(t))$ of $X(t)$ with the image $g(Y(t))$ of $Y(t)$.

Definition 3: *We say that* $\{f(X(t)), t \geq 0\} \preceq_{\phi} \{g(Y(t)), t \geq 0\}$ *if*

$$f(X(0)) \preceq_{\phi} g(Y(0)) \Longrightarrow f(X(t)) \preceq_{\phi} g(Y(t)), \forall t > 0.$$

If $\preceq_{\phi}$ represents the strong stochastic ordering $\preceq_{st}$, then we use the coupling theorem [2], [3], [16] to prove the stochastic comparison of Markov processes:

Theorem 2: The following propositions are equivalents:

- $\{f(X(t)), t \geq 0\} \preceq_{st} \{g(Y(t)), t \geq 0\}$.
- There exists a Markov process $\{Z(t) = (X'(t), Y'(t)), t \geq 0\}$, where $X'(t)$ and $Y'(t)$ have the same probability distributions then $X(t)$ and $Y(t)$, and $Z(t)$ has its values on $K = \{(x, y) \in E \times F, f(x) \preceq g(y)\}$.

REFERENCES

[1] T. S. Rappaport, *Wireless Communications*. Prentice Hall, 2001.

[2] M. Doisy, "Comparaison de processus Markoviens," Ph.D. thesis, Université de Pau et des pays de l'Adour, 1992.

[3] D. Stoyan, *Comparaison Methods for Queues and Other Stochastics Models*. John Wiley and Sons, Inc., 1976.

[4] W. A. Massey, "Stochastic orderings for Markov processses on partially ordered spaces," *Mathematics of Operations Research*, vol. 12, pp. 350–367, 1987.

[5] J.-M. Vincent, "Encadrement de performances: Une approche algorithmique," *Rapport interne LGI-IMAG*, 1993.

[6] L. Truffet, "Méthodes de calcul de bornes stochastiques sur des modèles de systèmes et de réseaux," Ph.D. thesis Université Paris 6, 1995.

[7] J. M. Fourneau, L. Mokdad, and N. Pekergin, "Bounding the loss rates in a multistage ATM switch," in *Proc. 9th Int. Conf. Modelling Techniques and Tools for Computer Performance Evaluation*, France, 1997.

[8] J. M. Fourneau, N. Pekergin, and H. Taleb, "Stochastic bounds and QoS: Application to the loss rates in ATM networks," in *Proc. European Simulation Multiconference*, Budapest, 1996.

[9] H. Castel and L. Mokdad, "Performance measure bounds in mobile networks by state space reduction," in *Proc. the 13th IEEE/ACM MASCOTS*, Sept. 2005, pp. 397–406.

[10] J. Ben-Othman, H. Castel-Taleb, and L. Mokdad, "Performance evaluation of mobile networks based on stochastic ordering of Markov chains," *Studia Informatica Universalis*, vol. 4, no. 2, pp. 191–208, 2006.

[11] S. Sousa and J. A. Silvester, "Spreading code protocols for distributed spreading-spectrm packet radio networks" *IEEE Trans. Commun.*, vol. 36, no. 3, pp. 776–783, 1998.

[12] I. Teerawat, E. Hossain, and I. K. Dong, "Medium access control protocols for wireless mobile ad hoc networks: Issues and approaches," *Wireless Commun. Mob. Comput.*, vol. 3, pp. 935–958, 2003.

[13] M. Sidi and D. Starobinski, "New call blocking versus handoff blocking in cellular networks," *Baltzer J.*

[14] G. Hebuterne and A. Gravey, "A space priority queueing mechanism for multiplexing ATM channels," *ITC Specialist Seminar, Computer Network and ISDN Systems*, vol. 20, pp. 37–42, 1990.

[15] W. K. Grassman, M. I. Taksar, and D. P. Heyman, "Regenerative analysis and steady state distributions for Markov chains," *Oper. Res. J.*, vol. 13, no. 3, pp. 1107–1116.

[16] T. Lindvall, *Lectures on the Coupling Method.* John Wiley and Sons, Inc., 1992.

Jalel Ben-Othman received his B.Sc. and M.Sc. degrees both in Computer Science from the University of Pierre et Marie Curie, France in 1992 and 1994, respectively. He received a Ph.D. degree from University of Versailles, France, in 1998. He was at the University of Orsay (Paris 11) and University of Pierre et Marie Curie (Paris 6), in 1998 and 1999 respectively, where he was an Assistant Professor. He joined the Department of Computer Science at the University of Versailles in 2000, where he is now an Associate Professor. His research interests are in the area of Broadband Wireless Networks, with a particular emphasis on multi-services bandwidth management in WLAN (IEEE 802.11), WMAN (IEEE 802.16), WWAN (LTE), and ad-hoc sensor networks. He is also interested in security in sensor and ad-hoc networks. He has been on the technical program committees of more than 20 international conferences and workshops. He is involved in different IEEE technical committees, especially in AHSN. He is a Member of IEEE and ACM.

Hind Castel-Taleb received the Ph.D. in computer science from the University of Paris 6 in 1996. Currently, she is an Assistant Professor at the INT (Institut National des Telecommunications). Her main research interests are about traffic, QoS in optical networks and mobile networks and mathematical methods as stochastic comparisons for the performance evaluation of networks.

Lynda Mokdad received the Ph.D. in computer science from the University of Versailles in 1997. She is an Assistant Professor at Lamsade, university Paris-Dauphine. Her main research interests are about performance evaluation techniques and applications in wired, mobile, and wireless networks and in software technologies as Web services.